Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

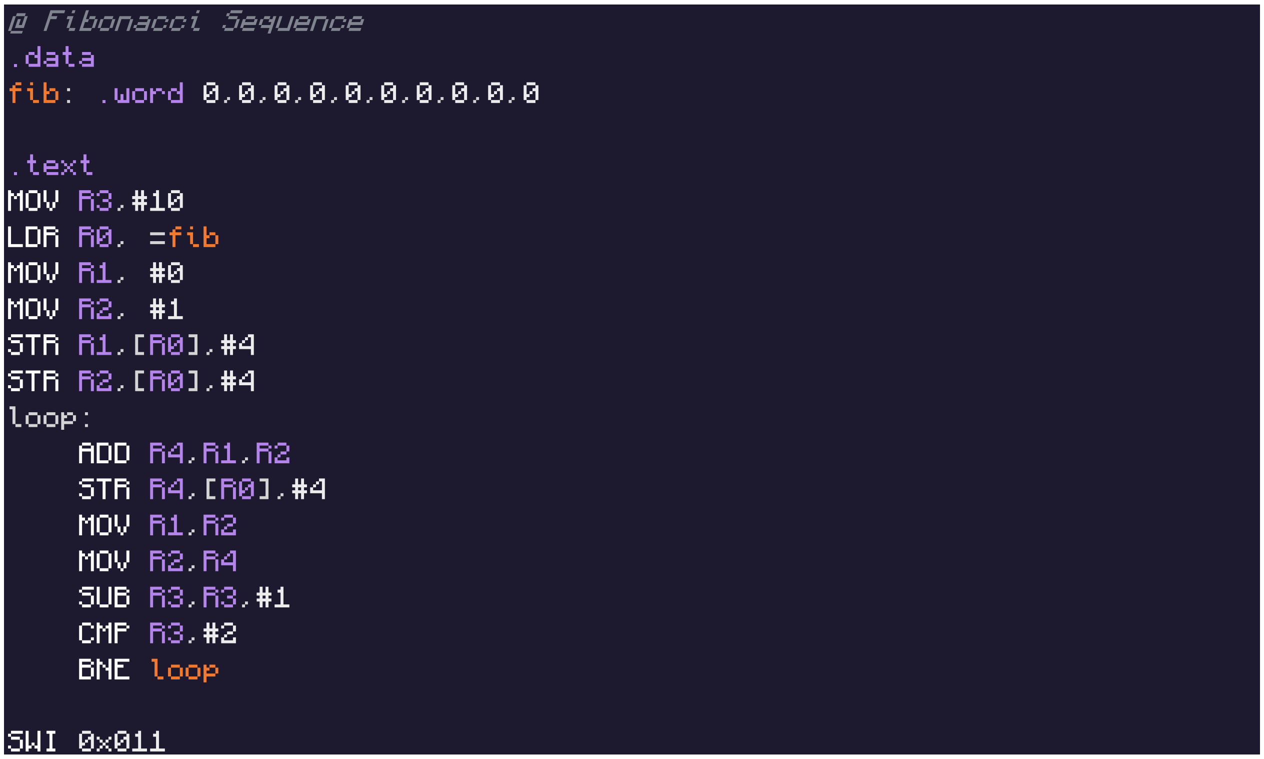
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____1__

Title of the Program

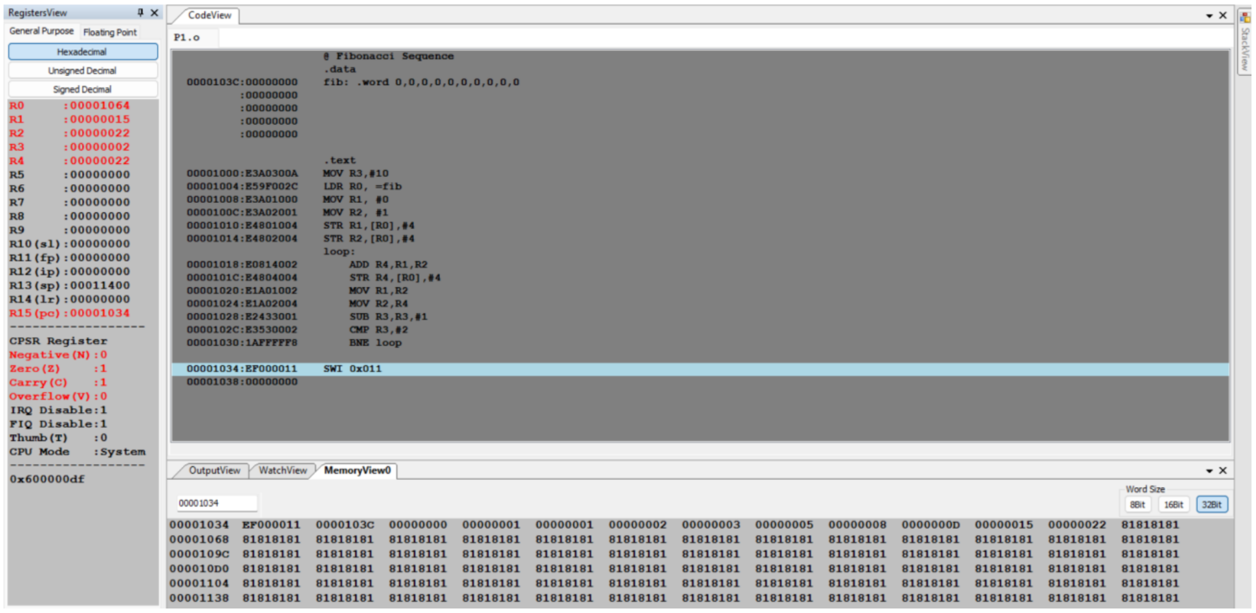
Generate Fibonacci Series and store them in an array.

I.ARM Assembly Code:



```
@ Fibonacci Sequence
.data
fib: .word 0,0,0,0,0,0,0,0,0,0

.text
MOV R3,#10
LDR R0, =fib
MOV R1, #0
MOV R2, #1
STR R1,[R0],#4
STR R2,[R0],#4
loop:
    ADD R4,R1,R2
    STR R4,[R0],#4
    MOV R1,R2
    MOV R2,R4
    SUB R3,R3,#1
    CMP R3,#2
    BNE loop

SWI 0x011
```

II. Output Screen Shots (One):



```
R0        :00001064
R1        :00000015
R2        :00000022
R3        :00000002
R4        :00000022
R5        :00000000
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):00001034
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
                        @ Fibonacci Sequence
                        .data
0000103C:00000000       fib: .word 0,0,0,0,0,0,0,0,0,0
        :00000000
        :00000000
        :00000000
        :00000000

                        .text
00001000:E3A0300A       MOV R3,#10
00001004:E59F002C       LDR R0, =fib
00001008:E3A01000       MOV R1, #0
0000100C:E3A02001       MOV R2, #1
00001010:E4801004       STR R1,[R0],#4
00001014:E4802004       STR R2,[R0],#4
                        loop:
00001018:E0814002           ADD R4,R1,R2
0000101C:E4804004           STR R4,[R0],#4
00001020:E1A01002           MOV R1,R2
00001024:E1A02004           MOV R2,R4
00001028:E2433001           SUB R3,R3,#1
0000102C:E3530002           CMP R3,#2
00001030:1AFFFFF8           BNE loop

00001034:EF000011       SWI 0x011
00001038:00000000
```

```
00001034

00001034 EF000011 0000103C 00000000 00000001 00000001 00000002 00000003 00000005 00000008 0000000D 00000015 00000022 81818181
00001068 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000109C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010D0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001104 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001138 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

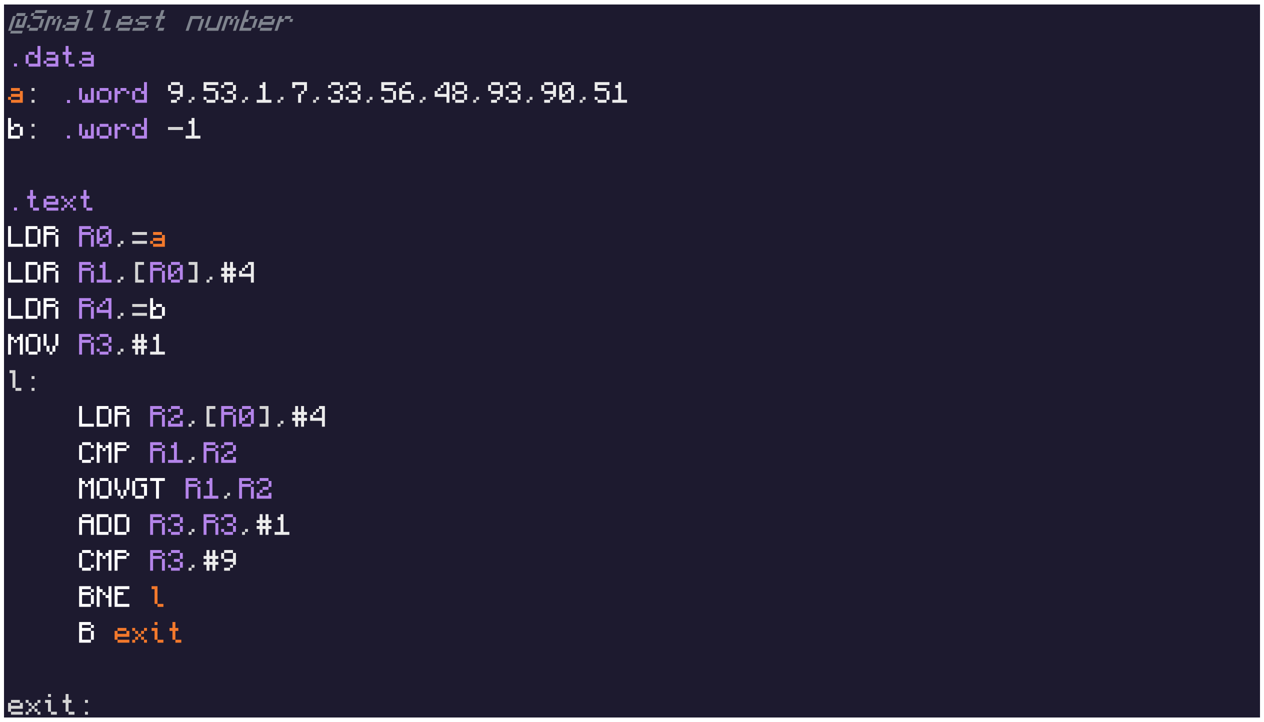
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____2___

Title of the Program

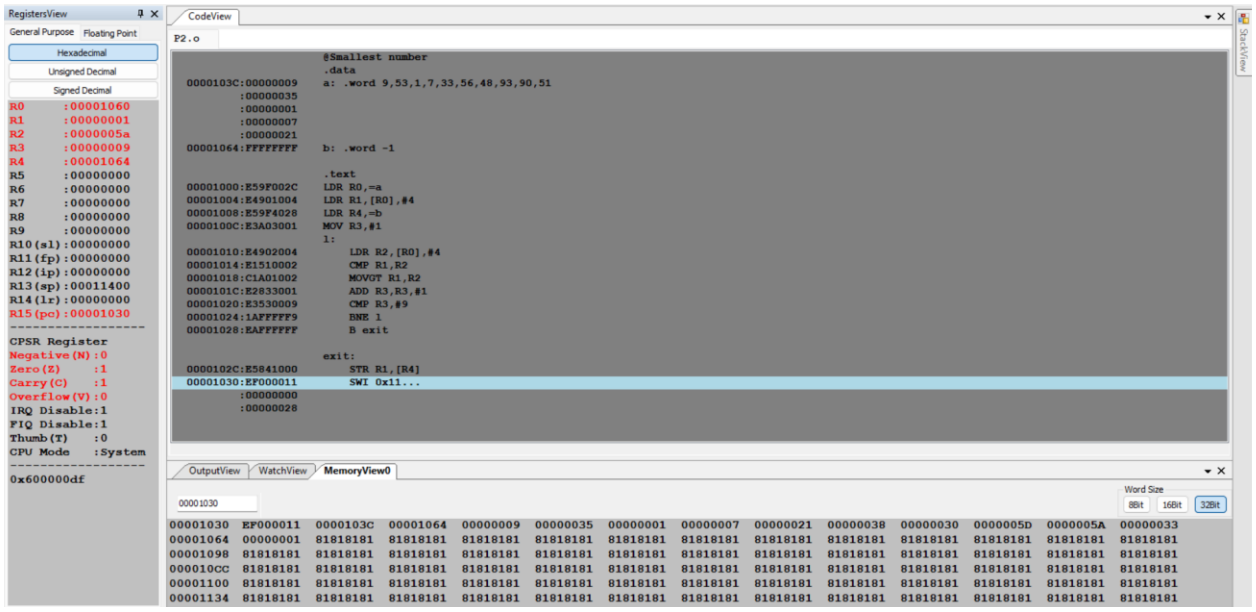
Write an ALP to find smallest number in an array of n 32-bit numbers

I.ARM Assembly Code:



```
@Smallest number
.data
a: .word 9,53,1,7,33,56,48,93,90,51
b: .word -1

.text
LDR R0,=a
LDR R1,[R0],#4
LDR R4,=b
MOV R3,#1
l:
    LDR R2,[R0],#4
    CMP R1,R2
    MOVGT R1,R2
    ADD R3,R3,#1
    CMP R3,#9
    BNE l
    B exit

exit:
    STR R1,[R4]
    SWI 0x11
```

II. Output Screen Shots (One):



```
R0       :00001060
R1       :00000001
R2       :0000005a
R3       :00000009
R4       :00001064
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00000000
R15(pc)  :00001030
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
                        @Smallest number
                        .data
0000103C:00000009       a: .word 9,53,1,7,33,56,48,93,90,51
        :00000035
        :00000001
        :00000007
        :00000021
00001064:FFFFFFFF       b: .word -1

                        .text
00001000:E59F002C       LDR R0,=a
00001004:E4901004       LDR R1,[R0],#4
00001008:E59F4028       LDR R4,=b
0000100C:E3A03001       MOV R3,#1
                        l:
00001010:E4902004           LDR R2,[R0],#4
00001014:E1510002           CMP R1,R2
00001018:C1A01002           MOVGT R1,R2
0000101C:E2833001           ADD R3,R3,#1
00001020:E3530009           CMP R3,#9
00001024:1AFFFFF9           BNE l
00001028:EAFFFFFF           B exit

                        exit:
0000102C:E5841000           STR R1,[R4]
00001030:EF000011           SWI 0x11...
        :00000000
        :00000028
```

```
00001030

00001030  EF000011  0000103C  00001064  00000009  00000035  00000001  00000007  00000021  00000038  00000030  0000005D  0000005A  00000033
00001064  00000001  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001098  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010CC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001100  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001134  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

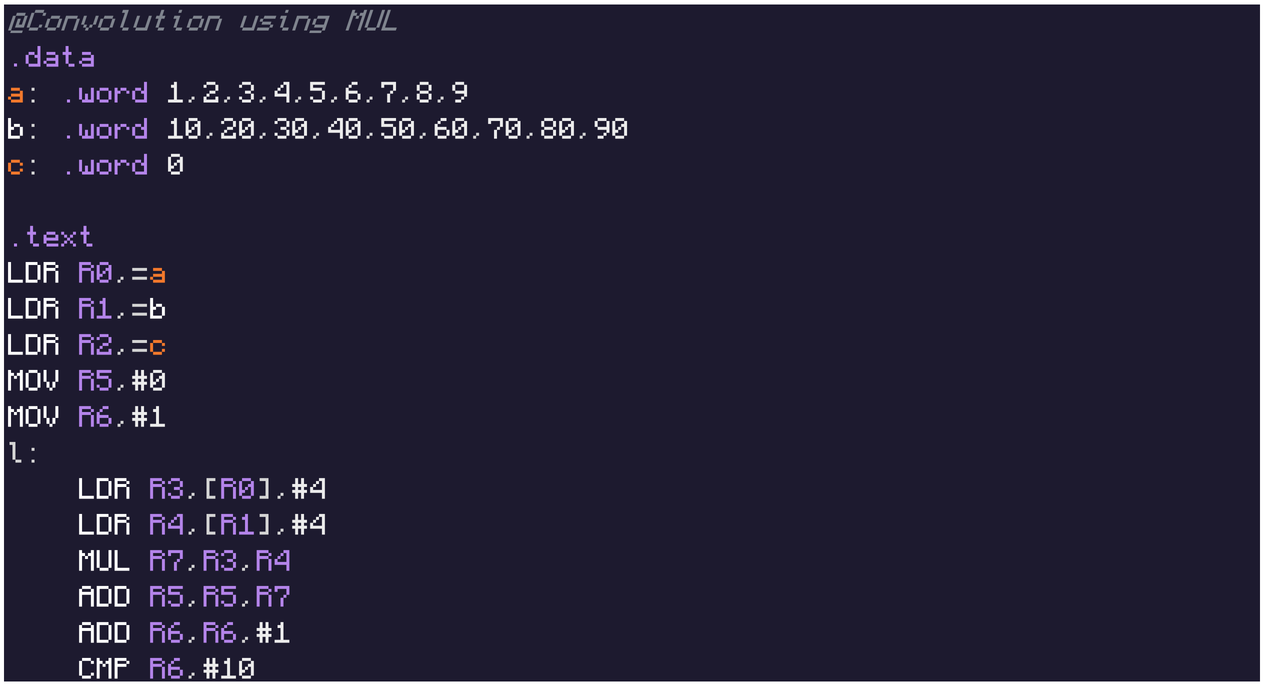
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____3__

Title of the Program

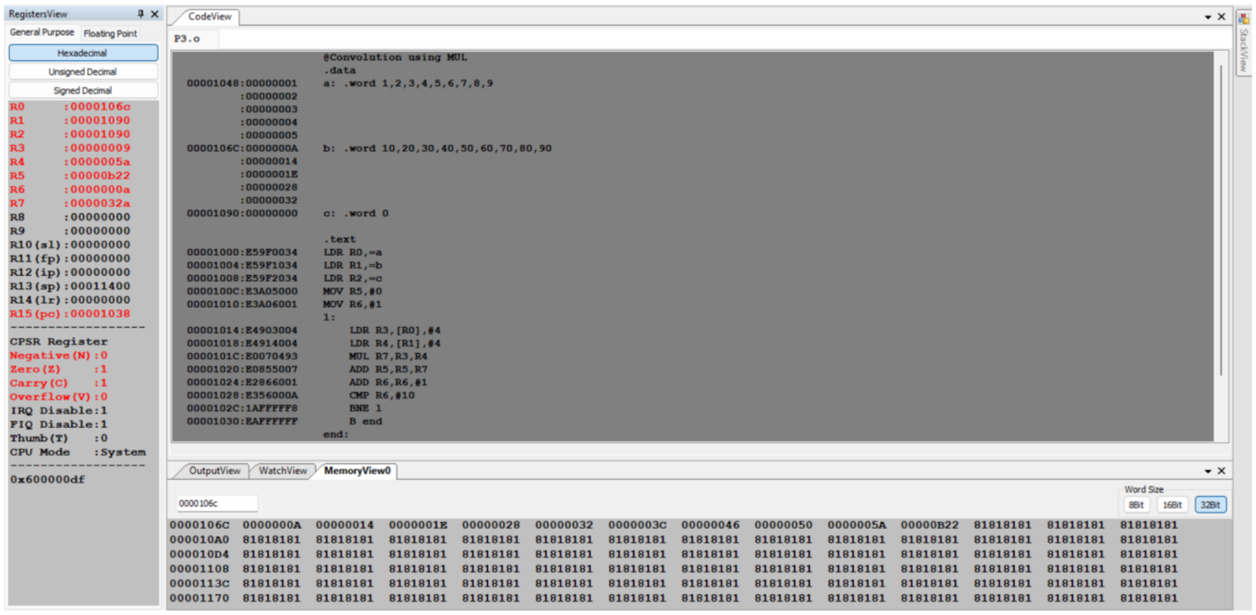
To perform Convolution using MUL instruction (Addition of multiplication of respective numbers of loc A and loc B)

I.ARM Assembly Code:



```
@Convolution using MUL
.data
a: .word 1,2,3,4,5,6,7,8,9
b: .word 10,20,30,40,50,60,70,80,90
c: .word 0

.text
LDR R0,=a
LDR R1,=b
LDR R2,=c
MOV R5,#0
MOV R6,#1
l:
    LDR R3,[R0],#4
    LDR R4,[R1],#4
    MUL R7,R3,R4
    ADD R5,R5,R7
    ADD R6,R6,#1
    CMP R6,#10
    BNE l
    B end
end:
    STR R5,[R2]
    SWI 0x11
```

II. Output Screen Shot (One):



```
R0        :0000106c
R1        :00001090
R2        :00001090
R3        :00000009
R4        :0000005a
R5        :00000b22
R6        :0000000a
R7        :0000032a
R8        :00000000
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00011400
R14(lr)   :00000000
R15(pc)   :00001038
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
                            @Convolution using MUL
                            .data
00001048:00000001           a: .word 1,2,3,4,5,6,7,8,9
        :00000002
        :00000003
        :00000004
        :00000005
0000106C:0000000A           b: .word 10,20,30,40,50,60,70,80,90
        :00000014
        :0000001E
        :00000028
        :00000032
00001090:00000000           c: .word 0

                            .text
00001000:E59F0034           LDR R0,=a
00001004:E59F1034           LDR R1,=b
00001008:E59F2034           LDR R2,=c
0000100C:E3A05000           MOV R5,#0
00001010:E3A06001           MOV R6,#1
                            l:
00001014:E4903004               LDR R3,[R0],#4
00001018:E4914004               LDR R4,[R1],#4
0000101C:E0070493               MUL R7,R3,R4
00001020:E0855007               ADD R5,R5,R7
00001024:E2866001               ADD R6,R6,#1
00001028:E356000A               CMP R6,#10
0000102C:1AFFFFF8               BNE l
00001030:EAFFFFFF               B end
                            end:
```

```
0000106c
0000106C  0000000A  00000014  0000001E  00000028  00000032  0000003C  00000046  00000050  0000005A  00000B22  81818181  81818181  81818181
000010A0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010D4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001108  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000113C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001170  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

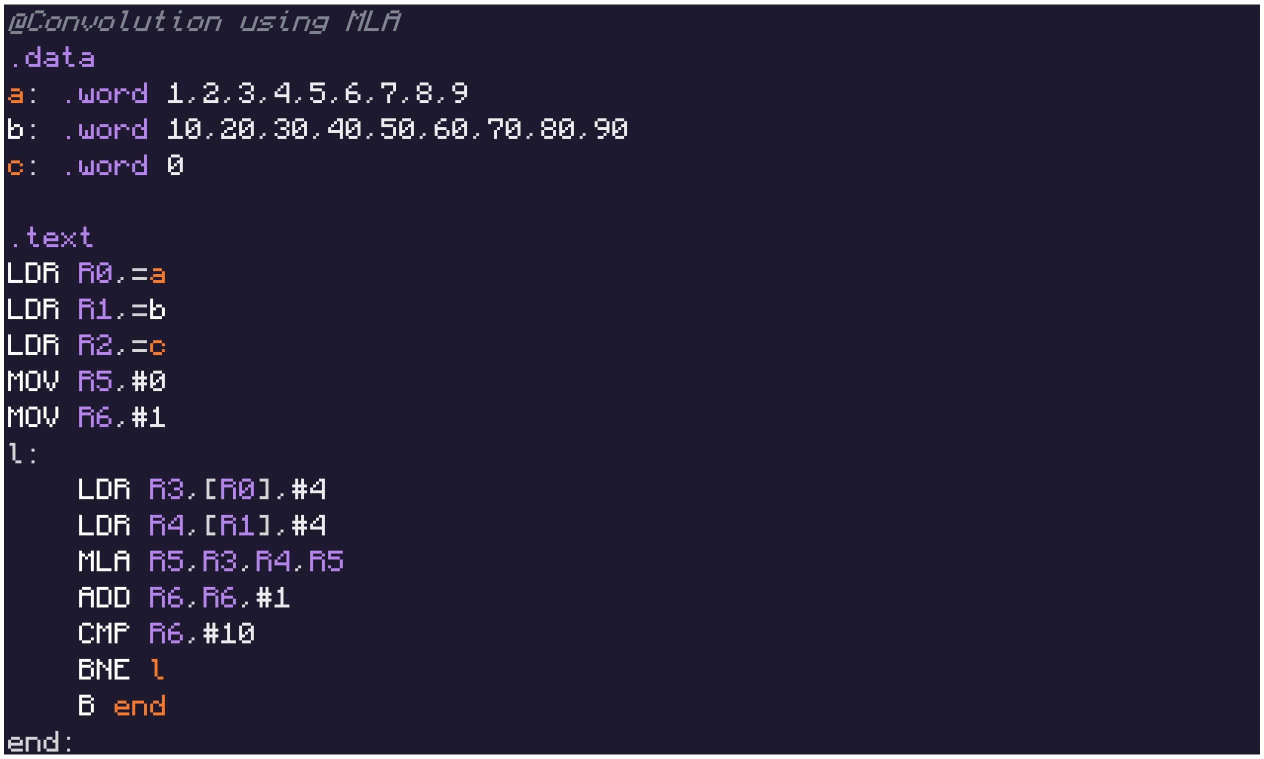
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____4___

Title of the Program

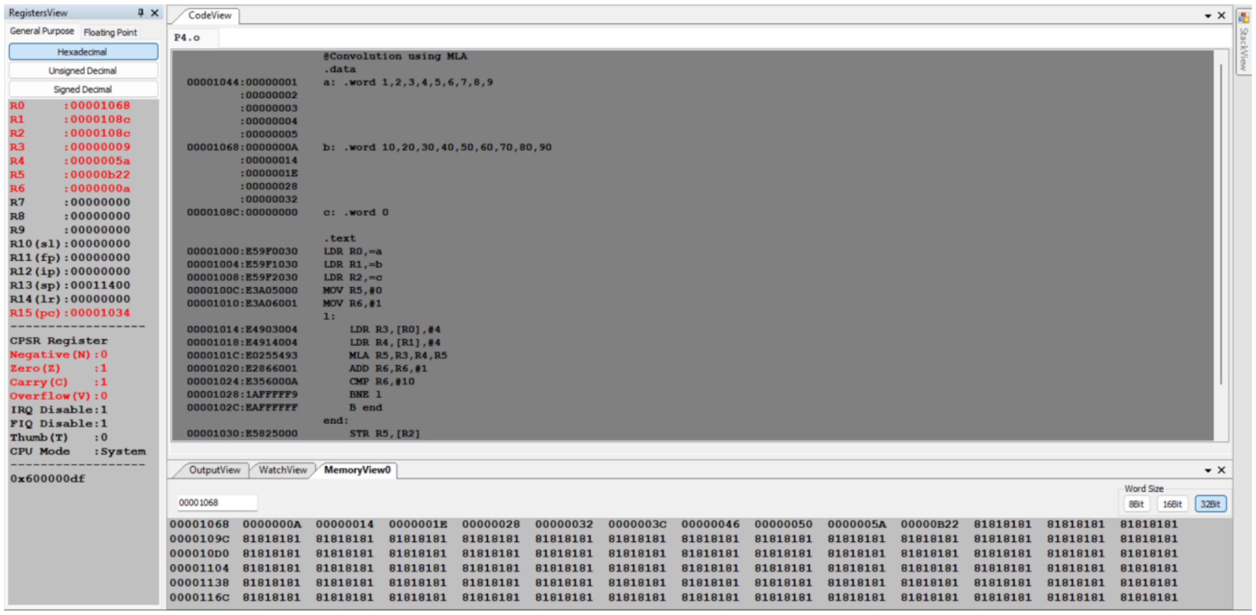
To perform Convolution using MLA instruction (Addition of multiplication of respective numbers of loc A and loc B).

I.ARM Assembly Code:



```
@Convolution using MLA
.data
a:  .word 1,2,3,4,5,6,7,8,9
b:  .word 10,20,30,40,50,60,70,80,90
c:  .word 0

.text
LDR R0,=a
LDR R1,=b
LDR R2,=c
MOV R5,#0
MOV R6,#1
l:
    LDR R3,[R0],#4
    LDR R4,[R1],#4
    MLA R5,R3,R4,R5
    ADD R6,R6,#1
    CMP R6,#10
    BNE l
    B end
end:
    STR R5,[R2]
    SWI 0x11
```

II. Output Screen Shot (One):



```
R0       :00001068
R1       :0000108c
R2       :0000108c
R3       :00000009
R4       :0000005a
R5       :00000b22
R6       :0000000a
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00000000
R15(pc)  :00001034
------------------
CPSR Register
Negative(N) :0
Zero(Z)     :1
Carry(C)    :1
Overflow(V) :0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
P4.o
                        @Convolution using MLA
                        .data
00001044:00000001       a: .word 1,2,3,4,5,6,7,8,9
        :00000002
        :00000003
        :00000004
        :00000005
00001068:0000000A       b: .word 10,20,30,40,50,60,70,80,90
        :00000014
        :0000001E
        :00000028
        :00000032
0000108C:00000000       c: .word 0

                        .text
00001000:E59F0030       LDR R0,=a
00001004:E59F1030       LDR R1,=b
00001008:E59F2030       LDR R2,=c
0000100C:E3A05000       MOV R5,#0
00001010:E3A06001       MOV R6,#1
                        l:
00001014:E4903004           LDR R3,[R0],#4
00001018:E4914004           LDR R4,[R1],#4
0000101C:E0255493           MLA R5,R3,R4,R5
00001020:E2866001           ADD R6,R6,#1
00001024:E356000A           CMP R6,#10
00001028:1AFFFFF9           BNE l
0000102C:EAFFFFFF           B end
                        end:
00001030:E5825000           STR R5,[R2]
```

```
00001068
00001068  0000000A  00000014  0000001E  00000028  00000032  0000003C  00000046  00000050  0000005A  00000B22  81818181  81818181  81818181
0000109C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010D0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001104  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001138  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000116C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

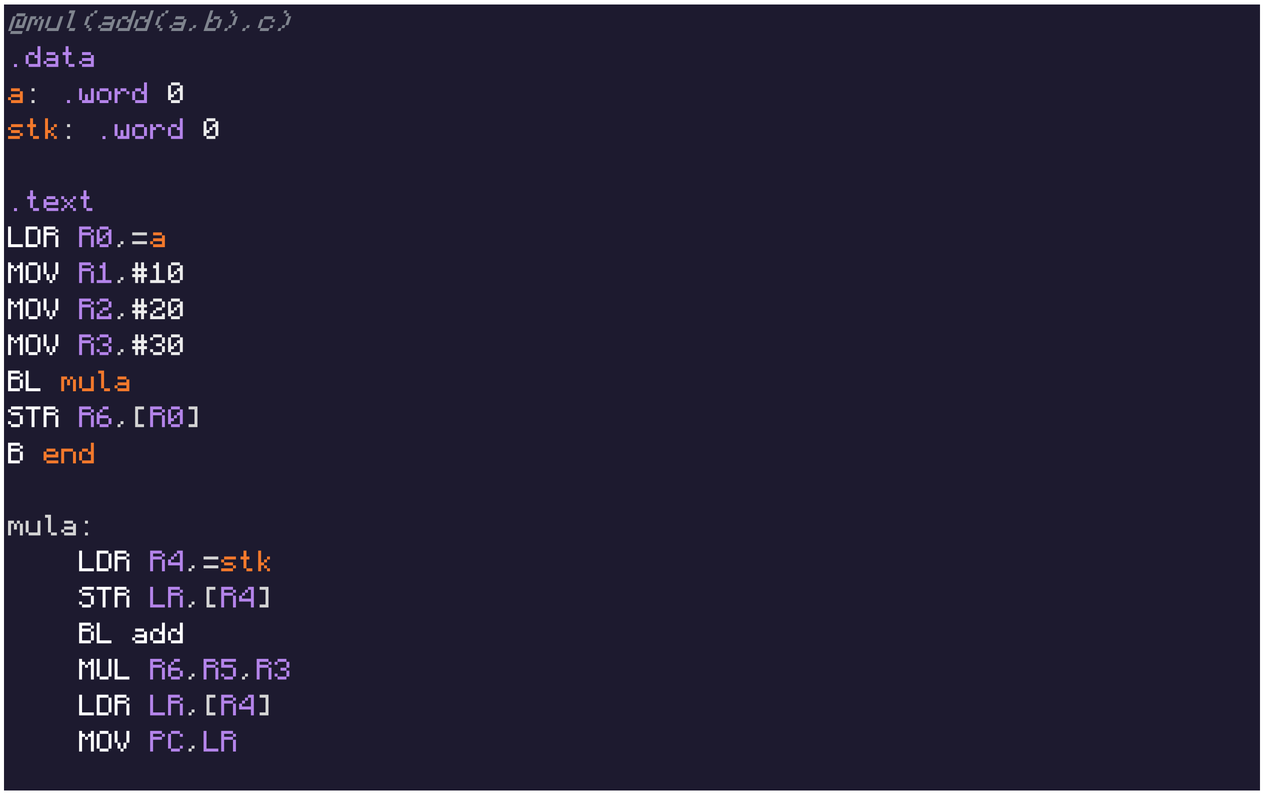
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____5__

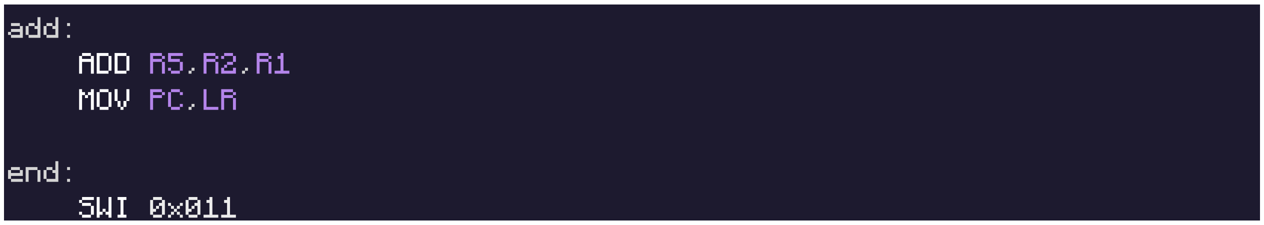
Title of the Program

Write an ALP to find mul(add(a,b),c)

I.ARM Assembly Code:

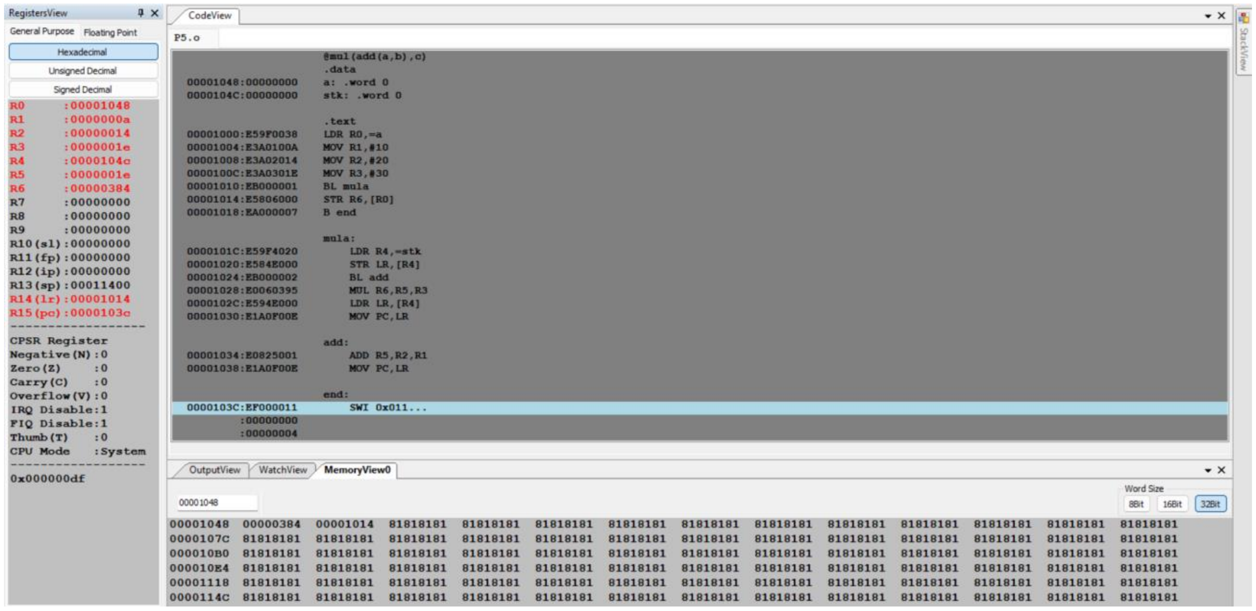


```
@mul(add(a,b),c)
.data
a: .word 0
stk: .word 0

.text
LDR R0,=a
MOV R1,#10
MOV R2,#20
MOV R3,#30
BL mula
STR R6,[R0]
B end

mula:
    LDR R4,=stk
    STR LR,[R4]
    BL add
    MUL R6,R5,R3
    LDR LR,[R4]
    MOV PC,LR
```



```
add:
    ADD R5,R2,R1
    MOV PC,LR

end:
    SWI 0x011
```

II. Output Screen Shot (One):



```
R0       :00001048
R1       :0000000a
R2       :00000014
R3       :0000001e
R4       :0000104c
R5       :0000001e
R6       :00000384
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00001014
R15(pc)  :0000103c
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x000000df
```

```
                      @mul(add(a,b),c)
                      .data
00001048:00000000     a: .word 0
0000104C:00000000     stk: .word 0

                      .text
00001000:E59F0038     LDR R0,=a
00001004:E3A0100A     MOV R1,#10
00001008:E3A02014     MOV R2,#20
0000100C:E3A0301E     MOV R3,#30
00001010:EB000001     BL mula
00001014:E5806000     STR R6,[R0]
00001018:EA000007     B end

                      mula:
0000101C:E59F4020         LDR R4,=stk
00001020:E584E000         STR LR,[R4]
00001024:EB000002         BL add
00001028:E0060395         MUL R6,R5,R3
0000102C:E594E000         LDR LR,[R4]
00001030:E1A0F00E         MOV PC,LR

                      add:
00001034:E0825001         ADD R5,R2,R1
00001038:E1A0F00E         MOV PC,LR

                      end:
0000103C:EF000011         SWI 0x011...
         :00000000
         :00000004
```

```
00001048
00001048  00000384  00001014  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000107C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010E4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001118  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000114C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

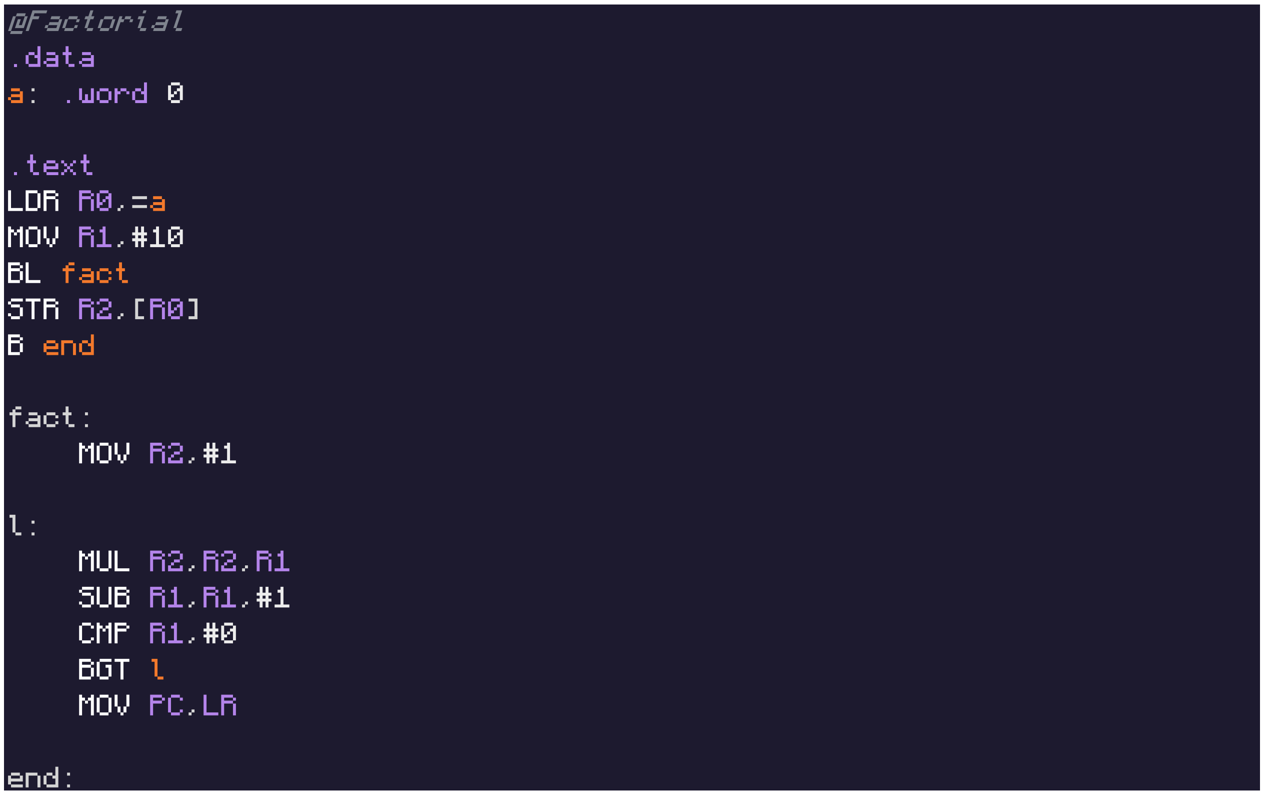
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____6__

Title of the Program

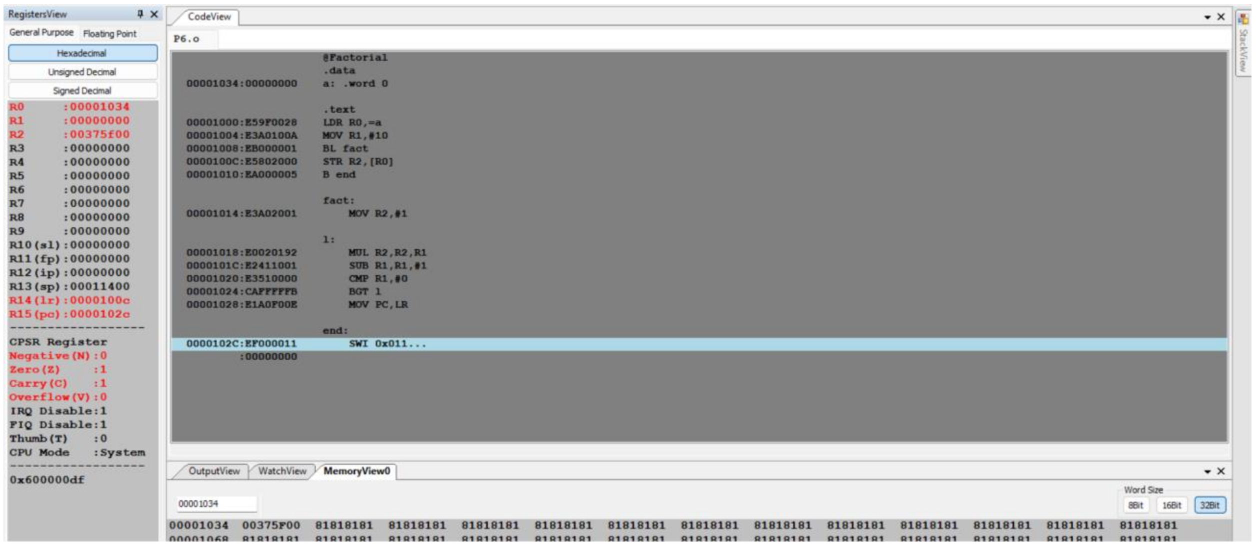
Write an ALP to find factorial using subroutine

I.ARM Assembly Code:



```
@Factorial
.data
a: .word 0

.text
LDR R0,=a
MOV R1,#10
BL fact
STR R2,[R0]
B end

fact:
    MOV R2,#1

l:
    MUL R2,R2,R1
    SUB R1,R1,#1
    CMP R1,#0
    BGT l
    MOV PC,LR

end:
    SWI 0x011
```

II. Output Screen Shot (One):



```
R0        :00001034
R1        :00000000
R2        :00375f00
R3        :00000000
R4        :00000000
R5        :00000000
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00011400
R14(lr)   :0000100c
R15(pc)   :0000102c
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
P6.o
                        @Factorial
                        .data
00001034:00000000       a: .word 0

                        .text
00001000:E59F0028       LDR R0,=a
00001004:E3A0100A       MOV R1,#10
00001008:EB000001       BL fact
0000100C:E5802000       STR R2,[R0]
00001010:EA000005       B end

                        fact:
00001014:E3A02001           MOV R2,#1

                        l:
00001018:E0020192           MUL R2,R2,R1
0000101C:E2411001           SUB R1,R1,#1
00001020:E3510000           CMP R1,#0
00001024:CAFFFFFB           BGT l
00001028:E1A0F00E           MOV PC,LR

                        end:
0000102C:EF000011           SWI 0x011...
        :00000000
```

```
00001034
00001034  00375F00  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001068  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 08/02/23

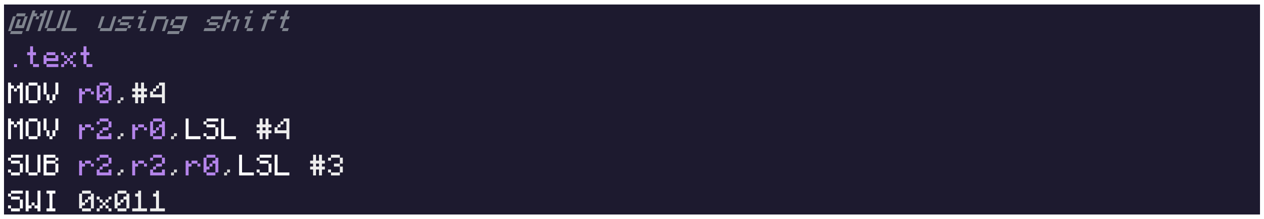
Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F

Week#____3_______ Program Number: ____7__

Title of the Program

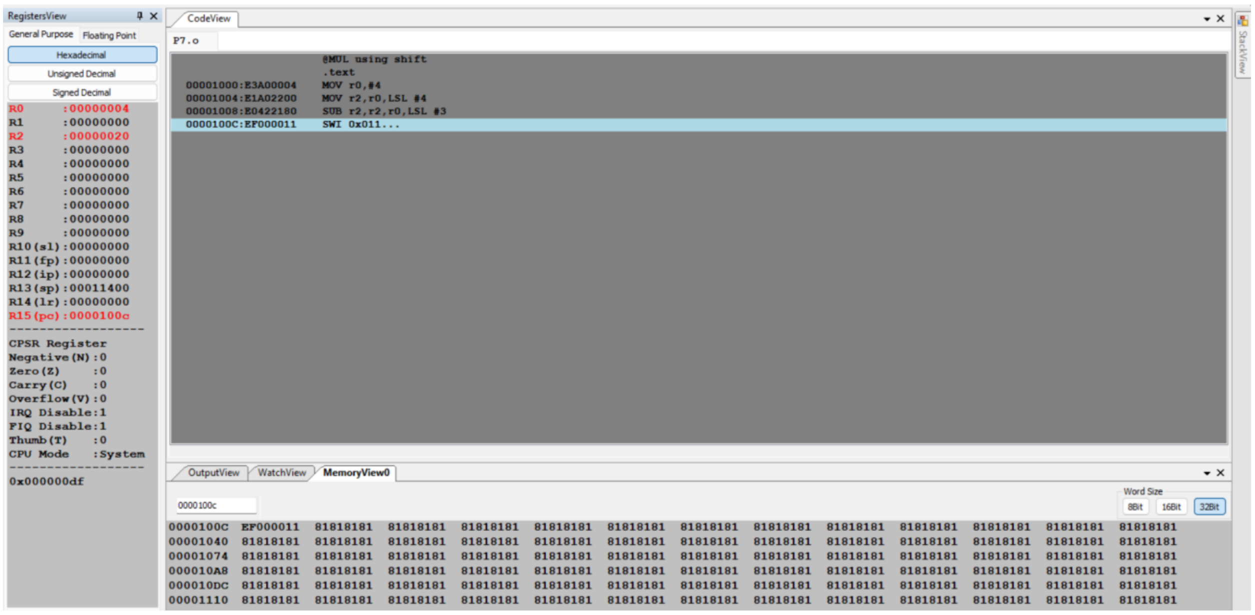
Write an ALP to perform multiplication using shift method (without using MUL)

I.ARM Assembly Code:



```
@MUL using shift
.text
MOV r0,#4
MOV r2,r0,LSL #4
SUB r2,r2,r0,LSL #3
SWI 0x011
```

II. Output Screen Shot (One):



```
R0        :00000004
R1        :00000000
R2        :00000020
R3        :00000000
R4        :00000000
R5        :00000000
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00011400
R14(lr)   :00000000
R15(pc)   :0000100c
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x000000df
```

```
                        @MUL using shift
                        .text
00001000:E3A00004       MOV r0,#4
00001004:E1A02200       MOV r2,r0,LSL #4
00001008:E0422180       SUB r2,r2,r0,LSL #3
0000100C:EF000011       SWI 0x011...
```

```
0000100c

0000100C EF000011 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001040 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001074 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010A8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010DC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001110 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: Nikhil Girish
Name: Nikhil
SRN: PES2UG21CS334
Section: 4F
Date: 08/02/23